

Domain-specific Question Generation from a Knowledge Base

Linfeng Song¹ and Lin Zhao²

¹Computer Science Department, University of Rochester, Rochester, NY, 14623

²Bosch Research and Technology Center, Palo Alto, CA, 94304

Abstract

Question generation has been a research topic for a long time, where a big challenge is how to generate deep and natural questions. To tackle this challenge, we propose a system to generate natural language questions from a domain-specific knowledge base (KB) by utilizing rich web information. A small number of question templates are first created based on the KB and instantiated into questions, which are used as seed set and further expanded through the web to get more question candidates. A filtering model is then applied to select candidates with high grammaticality and domain relevance. The system is able to generate large amount of in-domain natural language questions with considerable semantic diversity and is easily applicable to other domains. We evaluate the quality of the generated questions by human judgments and the results show the effectiveness of our proposed system.

1 Introduction

Questions are useful for student assessment or coaching purpose in educational or professional contexts, such as student’s self-assessment or new employee training about products or industrial procedures. A large-scale question corpora is also critical to many NLP tasks including question answering, dialogue interaction and intelligent tutoring system, where large amount of annotated questions are usually needed as training data in supervised learning. Until now most of the applications rely on manual data collection which typically takes a lot of time and efforts. Although crowd-sourcing is becoming more and more popular for data collection, it is sometimes hard to

ensure the qualities of the data, especially if the worker is not an expert in the related fields for a domain-specific task. Not to say the collection has to be conducted every time when there comes a new domain or the existing domain gets updated.

Automatic question generation has long been of interest to the research community. Most work is focusing on generating questions from a text (Curto et al., 2012; Olney et al., 2012; Mazidi and Nielsen, 2014; Labutov et al., 2015). With the rising of structured knowledge base (KB), some work starts to utilize KB to generate questions (Seyler et al., 2015; Serban et al., 2016). KB can contain knowledge in either open domain such as Freebase (Bollacker et al., 2008) and DBpedia (Auer et al., 2007) or closed domain such as those available in many industrial domains (e.g., product KB, medical KB).

Our work is in this line of research where we tackle the problem of question generation from domain specific KB. The limitations with existing work are that the generated questions are generally simple questions with limited scope, or a lot of manual work has to be involved to hand-craft all the question templates, making the adaptation to a new domain costly.

People ask questions on the web every day, which generally represent the users’ information needs and naturally form a huge resource of question corpora. In our approach, we try to minimize the manual effort in question generation by leveraging the web resource.

Given a KB specific to a certain domain, We start with a small set of hand-crafted templates based on the relationship defined in the KB. These templates include placeholders to replace the string of the subject or object. A seed question set is then generated from the templates with concepts filled in. This seed set is further expanded through the web, by sending each seed question

to a search engine as a search query to retrieve more related question candidates. Finally a filter is applied to estimate the fluency of each candidate based on language modeling, and also its relevance to the domain based on distributional semantic similarity calculation, and only those with fluency or relevance score above a threshold are returned. We hypothesize that the resulting questions are both natural and relevant to the working domain.

In our framework, the manual effort is involved in only seed template construction, where we do not require an extensive number of templates to be created to cover as many as possible the different ways of forming a question. Only one or a few most common question templates is needed as seeds for each of the essential relations in the KB, so as to reduce the human efforts, with the rest of the framework fully automatic. By leveraging the huge amount of available web data, we are able to obtain more natural and open-ended questions more close to the users' real information needs. This framework can be easily applied to other KB to effectively generate questions for a new domain or application.

To summarize, the main contributions of our work are:

- We propose a new framework of KB-based question generation for domain-specific applications, which is applicable to any KB or domain.
- We propose a mechanism to build a large-scale question set with significantly reduced human efforts by exploiting the substantial web content which yields more diverse and semantically richer questions. By taking into consideration of whether the question is grammatical and whether it is relevant to the working domain, the resulting questions will be both natural and relevant.
- We evaluate the system with human judgments on the constructed question set, and show that it is effective.

2 Related Work

Automatic question generation has received increasing interest from the Natural Language Generation (NLG) community, been further advanced by the Question Generation Shared Task and Evaluation Challenge (QGSTEC) (Rus et al., 2010),

which created a common corpus for empirical evaluation of question generation.

Base on the type of input, question generation task can be divided into two categories: text-based and KB-based. Question generation from text has been relatively well studied where the input can be either single or multiple sentences.

The majority of existing systems generate questions from a single sentence. The source sentence is analyzed and its portions are selected as the content of the question (i.e., what to ask), then transformation rules are constructed to transform the source sentence into a valid question (i.e., how to ask). Among the work, some utilizes syntactic parsing to identify the targets of questions and construct syntactic transformation rules to generate questions (Wolfe, 1976; Ali et al., 2010; Heilman and Smith, 2010; Curto et al., 2012). Some utilizes semantic information (Chen et al., 2009; Lindberg et al., 2013; Mazidi and Nielsen, 2014) where semantic role labeling is applied to identify patterns in the source sentences for question generation. However, these work mainly focuses on the surface form of the sentence and question, and the scope of the generated questions is limited.

Some work attempts to generate deeper questions from documents. Agarwal et al. (2011) uses discourse connectives to generate questions from selected text segments for different question types. Olney et al. (2012) first converts the text into concept maps from which questions are generated. Labutov et al. (2015) generate deep open-ended questions from Wikipedia text, where they use crowd-sourcing to construct question templates based on the category of the Wikipedia page, and then apply question ranking to select the final question. Although these approaches can generate questions at a deeper conceptual level, they involve big amount of human effort to formulate questions by handcrafting either question templates or transformation rules.

On the other hand, question generation from KB receives less attention. Seyler et al. (2015) generates quiz questions from knowledge graphs, where for each target entity, a SPARQL query is generated as an intermediate representation and turned into a natural language question by a simple predefined template. Because of the fixed template used, the type of question is limited to quiz question. Serban et al. (2016) has constructed a corpus of 30M factoid question and an-

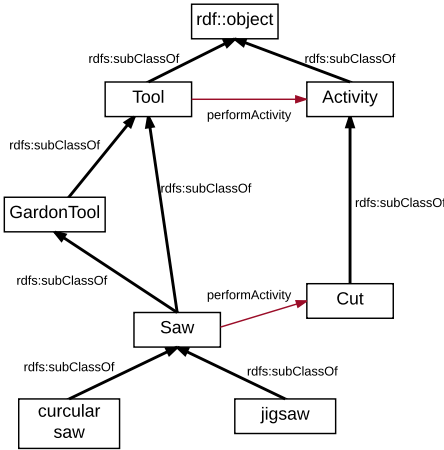


Figure 1: Example of depth definition.

swer pairs by training a recurrent neural network to map KB facts into corresponding natural language questions. However, their approach needs large amount of fact-question pairs as training data which is not necessarily available for each domain. Also the trained model only works for a single KB fact, which restricts the scope of the generated questions. Our work follows this direction as we believe KB is a good resource for data generation, especially for closed domains. The main difference with the existing work is that first our approach is unsupervised without requiring any labeled data, and second, the types of the generated questions are more diverse without any restriction.

3 Domain-specific Knowledge Base

A knowledge base (KB) is used to store knowledge which is typically represented using RDF, RDFS and OWL, the W3C standard for semantic web. A RDF KB can be seen as a graph, in which the nodes are entities and the labeled edges represent the relationships between the entities. The graph can also be represented as a list of triples in the form of *(subject, predicate, object)*.

Large-scale KBs such as Freebase (Bollacker et al., 2008) and DBpedia (Auer et al., 2007) are very popular and widely used in many NLP applications. They contain millions of facts generally about person, location, organization and so on in the open domain. In our work, since we aim at generating corpus for a closed domain, we work on an in-house domain-specific KB in Power Tool domain. This domain describes knowledge about professional tools used in home or garden, etc. The representations of our KB include en-

tity concepts such as tools (e.g., jigsaw, screw-driver, cordless drill), accessories (e.g., drill bit, saw blade), materials (e.g., wood, timber, multi-plex), and activity concepts (e.g., sawing, drilling, screwing), together with their relationships. The main high level taxonomy is shown in Figure [add figure here].

Figure 1 shows a fragment of the KB, which can also be represented in the form of triples as follows:

```
(rdf:object, rdfs:subClassOf, Tool)
(rdf:object, rdfs:subClassOf, Activity)
(Tool, performActivity, Activity)
(Tool, rdfs:subClassOf, GardenTool)
(Tool, rdfs:subClassOf, Saw)
(GardenTool, rdfs:subClassOf, Saw)
(Activity, rdfs:subClassOf, Cut)
(Saw, performActivity, Cut)
```

where “rdf:object” and “rdfs:subClassOf” are pre-defined RDF object and label from the RDF community¹.

Here we give some definitions for further description. A *parent entity* is the one that the current entity connect to with “rdfs:subClassOf” edge. An entity may have multiple parent entities, such as “Saw” (in Figure 1). A *sibling entity* shares common all parent entities of the current entity. For example, “circular saw” is a sibling entity of “jigsaw” and vice versa.

4 Framework

Shown in Figure 2, the framework contains 4 sub-processes: KB progressing, question template construction, seed question generation and question expansion. Considering a KB as a list of triples, KB processing is to remove useless ones for question generation. Generally we remove two kinds of triples: one kind of triples contain abstract subjects or objects such as “(Tool, performsActivity, Activity)” where both “Tool” and “Activity” are abstract. The other kind of triples are for relation definition. For example, “(bearingDiameter, type, FunctionalProperty)” defines “bearingDiameter” as a type of “FunctionalProperty”. We remove these kinds of triples because no natural questions can be generated from them. We manually construct question templates for each predicate. For example, “Can I do #Y# with #X#?” is constructed for the predicate of “performsAc-

¹<https://www.w3.org/TR/rdf-schema/>

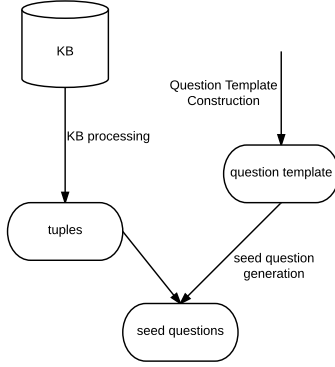


Figure 2: Overview of our framework.

tivity”. Combining that template with “(Jigsaw, performsActivity, CurveCut)”, we can generate a seed question “Can I do curve cut with jigsaw?”. Finally for question expansion, each seed question is thrown into Google search² for obtaining relevant questions.

4.1 KB Processing

Given a KB, we need to pre-process it before generating questions, because not every triple in it is suitable for question generation. Generally, we remove two kinds of triples: one kind contains abstract entities (subjects or objects). For example, “Tool” and “Activity” are abstract entities, but “Saw” and “cut” are not. It is easy for human to define, yet hard for computers to quantify. To quantify whether a given entity is abstract, we consider the graph with “`rdfs:subClassOf`” edges. We define the depth of entities that do not have an outgoing “`rdfs:subClassOf`” edge to be 0. Otherwise, the depth equals to the depth of the “shallowest parent” plus 1. Shown in Figure 1, the depth of “Saw” is 2, since the depth of its shallowest parent “Tool” is 1. We consider an entity is abstract if its depth is less than n . The other kind contains predicates such as “`rdfs:subClassOf`” and “`rdf:type`” that define the framework of the KB. We obtain the predicate list according to the RDF Schema³. As a result, this module outputs a list of triples which will be used to generate questions.

4.2 Question Template Construction

It is always a difficult problem to generate fluent and relevant questions from a KB, because the only input is a triple while the output is a fluent and

performsActivity	Can I perform #Y# with #X# How can I use #X# What activities can #X# perform
totalLength	How long is #X#
suitsMaterial	Is #Y# suitable for #X# What materials does #X# suit

Table 1: Example templates

relevant question. Recently Serban et al. (2016) introduces a method to generate 30M factoid questions from a KB. However, they use 100K human-crafted (triple, question) pairs to train their system, and the BLEU score (Papineni et al., 2002) of generated questions is only around 35.

Here we use a template-based method (in Section 4.3). For each predicate identified in the previous step, a few question templates are manually created. Shown in Table 1, for predicate “performActivity”, a template created is “What activities can #X# perform”, where #X# is the subject. Another template is “Can I perform #Y# with #Y#” where #X# is the subject and #Y# is the object. In this step, only a few (e.g., 2-3) representative templates are needed for a predicate, and the number of predicates are limited (tens of) as we deal with a domain-specific KB. The human effort is significantly lower than Serban et al. (2016).

4.3 Seed Question Generation

To generate seed questions, We use a two-step method which takes the constructed question templates and the triples as input. In the first step, for each template, a seed question set is generated by filling in the templates with values from associated triples. For example, for template “Can I perform #Y# with #X#?” and triple “(jigsaw, performsActivity, curve cut)”, we get the question “Can I perform curve cut with jigsaw?”. This process is repeated for each template so that all the variables are replaced with values.

In the second step, we replace a subject or object with a sibling class of the original entity to obtain a new question. A sibling class has the same parent class with the original class in the domain-specific KB. For example, for the question “Can I perform curve cut with jigsaw?”, we replace “jigsaw” with “circular saw” to result in a new question “Can I perform curve cut with circular saw?”. “circular saw” is a sibling class of “jigsaw” having the same parent class “saw”. We do not replace with an arbitrary class because people will not ask unreasonable questions such as “Can I per-

²<https://www.google.com/>

³<https://www.w3.org/TR/rdf-schema/>

form curve cut with hammer?”.

An advantage of our template-base method is that information about the subject, object and predicate are naturally present, which saves the human effort for annotation.

4.4 Question expansion

The seed questions are limited in terms of their scope and naturalness. To get more diversified and natural questions, we leverage web to expand our seed set. Each seed question in the seed question set is sent to a search engine to retrieve the related or suggested search queries provided by the search engine. These related queries can be again sent to the search engine to retrieve more queries. This process is iteratively performed until we get enough queries. The expansion method has the advantage that the expanded questions represent real users information needs.

The expanded questions may not be fluent or domain relevant, especially as the iteration go on, the domain relevance of newly expanded questions drops significantly. Previous methods either ignore this problem (Serban et al., 2016), or let human annotate training data to learn a classifier (Labutov et al., 2015). Our method does not rely on human effort by leveraging word embedding and language modeling. Taking the seed question set (section 4.3) as the in-domain data D_{in} , we filter out questions that are either ungrammatical or domain irrelevant. For domain relevance, we first calculate the embeddings of D_{in} and the candidate question, then discard the question if the cosine similarity between the two embeddings is lower than a threshold t . Shown in Equation 1, document embedding is calculated by averaging the word embeddings within it:

$$vec_d = \frac{\sum_{w \in d} vec_w}{len(d)} \quad (1)$$

where vec_w is the embedding for word w , $len(d)$ is the number of words in document d . Simple as it is, we show that it beats the state-of-the-art system on a domain relevance dataset in section 5.3. A question may only contains a few words resulting in data sparsity, we further enrich it with the snippets returned by searching it into a search engine.

For fluency, we use averaged language model score as Equation 2:

$$AvgLM(sent) = \frac{LM(sent)}{N} \quad (2)$$

how to change circular saw blade
how to measure lawn mower cutting height
how to sharpen drill bits on bench grinder
how does an oscillating multi tool work
how to cut a groove in wood without a router
what type of sander to use on deck
do i need a hammer drill
can i use acrylic paint on wood
how to use a sharpening stone with oil

Table 2: Example question expanded

where $LM(sent)$ is the language model score (log probability), and N is the word count of $sent$. This method has been shown effective on natural language generation (Liu and Zhang, 2015).

5 Experiments

5.1 Setup

We perform our main experiment with an in-house KB of power tool domain. It contains 67 different predicates, 293 different subjects and 279 different objects respectively. For the 67 predicates, we hand craft 163 templates, some examples are shown in Table 1. After KB processing we obtains 1498 triples, all of which are used for generating seed question set. After question expansion, we build dev and test sets to tune and test the fluency and domain relevance evaluation models. We first randomly select 1000 questions from the expanded question set for the development and test sets respectively, then ask 3 linguistic specialists to grade the fluency and domain relevance basing on a 3-point scheme, finally average the scores. A case is positive only if its score is greater than 2.

In addition to the main experiment, we also evaluate our framework on the 30M Factoid Question-Answer Corpus⁴, released by Serban et al. (2016). Each line in the dataset contains a triple (subject, predicate, object) and a question generated by their method. We randomly sample 10 lines, and compare our results with theirs.

To evaluate the fluency of the expanded questions, we train a 4-gram language model (LM) with KneserNey smoothing on gigaword (LDC2011T07), and learn word embeddings on Wikipedia⁵ for evaluating the relevance. We use the Skip-gram model (Mikolov and Dean, 2013) implementation from word2vec⁶ with default parameter setting to learn word embeddings. To en-

⁴<http://agarciaduran.org/>

⁵<https://dumps.wikimedia.org/>

⁶<https://code.google.com/archive/p/word2vec/>

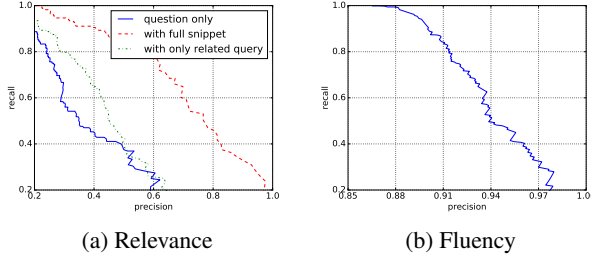


Figure 3: Curves between precision and recall on fluency and relevance evaluation

	Method	Dev	Test
Fluency	question	93.2	94.5
Relevance	question	44.6	34.1
	+related query	50.3	43.5
	+full snippet	68.8	59.9
All	question	43.0	32.8
	+related query	50.8	42.4
	+full snippet	68.2	58.9

Table 3: F-score of fluency and relevance evaluation models

rich questions with snippets, we try two ways: enrich with the *related queries* or with the *full snippet*, and compare their performances. Even though full snippet gives more context information, it also introduces more noise. On the devset we tune the thresholds of t_f (fluency) and t_r (relevance), which are further applied on the test set. We tune them both separately and jointly. For separate tuning, we show individual performances (Fluency and Relevance). For joint tuning, we perform grid search and show the overall performance (All).

We test our relevance evaluating method on the web snippet dataset, which is frequently used for domain classification of short documents. It contains 10,060 training and 2,280 test snippets (short documents) in 8 classes (domains), and there are 18 words in each snippet on average. There has been plenty of results (Phan et al., 2008; Chen et al., 2011; Ma et al., 2015) on the dataset. The fluency evaluation method has been shown competitive, especially when the test set is in a different domain (Liu and Zhang, 2015).

5.2 Results and Analysis

In the main experiment, we generated 12,228 seed questions from which 20,000 more questions are

Method	Precision
Phan et al. (2008)	82.18
Chen et al. (2011)	85.31
Ma et al. (2015)	85.48
question	85.65

Table 4: Precision on the web snippet dataset

expanded with Google search⁷. Shown in Table 2 are some expanded questions from which we can see that most of them are grammatical and relevant to the power tool domain. In addition, most questions are informative that correspond to a specific answer, except the one “do I need a hammer drill” that lacks some context. Finally, in addition to the simple factoid questions, our system generates many complex questions such as “how to cut a groove in wood without a router”.

We tried different values of t_f and t_r on the devsets, and show the curves between precision and recall in Figure 3. The dev and test results with tuned parameters are shown in Table 3. For relevance, enriching with full snippet is significantly better than the others, and enriching with related queries is significantly better than using only questions. This explains that the data sparseness is important for calculating document embedding. One example is the question “does saw 2 end” where “saw 2” actually refers to the movie “Saw II”. Its domain relatedness can not be determined only from its words, but it is much easier with the snippet. For fluency, the pick F-score is 93.3, and it only drops to around 88 if we keep all questions. This explains that the expanded questions are generally grammatical. In overall, the combined result does not drop too much which shows that there is positive correlation between the two indexes.

5.3 Domain Relevance

Shown in Table 4, We compare our question relevance evaluation method with previous state of the art methods: Phan et al. (2008) first derives latent topics with LDA (Blei et al., 2003) from a set of texts from Wikipedia, then uses the topics as appended features to expand the short text. Chen et al. (2011) further extend Phan et al. (2008) by using multi-granularity topics. Ma et al. (2015) adopts a Bayesian model that the probability a document D belongs to a topic t equals to the prior of t times the probability each word w in D comes

⁷<https://www.google.com/>

Ours	Serban et al. (2016)
what is the cultural heritage of churchill national park?	where in australia is churchill national park ?
what percentage of argentina’s population live in urban areas	what ’s one of the mountain where can you found in argentina in netflix ?
which country is the largest financial center of latin america	what is an organization that was born in latin america ?
which country has the largest freshwater lake in central america	what are the major town three gringos in venezuela and central america book ?
how does leukemia affect the body in children	who was someone who was involved in the leukemia ?
how does the nervous system maintain homeostasis	what is the drug category of central nervous system stimulation ?
why were colonial minutemen so prepared for the arrival of the british in concord	what county is concord in ?
which is the only country to have a bible on their national flag	whats the title of a book of the subject of the bible ?

Table 5: Example question expanded

from t . Our method (*question*) first calculate the document embedding for each test document and each domain in the training set, then assign test documents to the nearest (cosine similarity) domains. We do not enrich the short documents in this experiment.

Simple as it is, our method outperforms all previous methods without question enrichment, which proves the effectiveness of our method. One important reason is that word embeddings directly capture the similarity between distinct words, while it is hard for traditional methods. Besides, word embeddings are composable that sentence and document embeddings can be got by summing up the word embeddings within them. On the other hand, LDA only learns the similarity between words and topics, and it is trivial to determine a proper number of topics.

5.4 Comparison on 30M Factoid Question Answer Corpus

Serban et al. (2016) releases a corpus of 30M (triple, question) pairs from which we randomly select 500 triples to generate our questions. We first hand craft 53 templates, then generate 991 seed questions from the triples, finally get a expanded set of 1529 questions from Google search. We skip the KB-processing step as our input here are already triples. From the expanded question set, we select 500 by the averaged language model score as this is a domain-general corpus.

Shown in Table 5, we compare our questions with Serban et al. (2016) that questions in the same line describe the same entity. We can see that our questions are grammatical, natural as these questions are what people usually ask on the web. On the other hand, questions from Serban et al. (2016) are either ungrammatical (such as “who was someone who was involved in the leukemia ?” and “whats the title of a book of the subject of the bible ?”), unnatural (“what ’s one of the mountain where can you found in argentina in netflix ?”) or confusing (“who was someone who was involved in the leukemia ?”).

6 Conclusion

We presented an approach to generate natural language questions from knowledge graphs. By leveraging rich web information, the system is able to generate relevant questions with wide scope and the human effort can be significantly reduced. We evaluated our approach in terms of the grammaticality and relevance of the generated questions and the results showed its effectiveness. Note that although the work presented in this paper is for a specific domain, the working flow constitutes a general framework that could potentially be applied to other domains or KBs as well.

References

- [Agarwal et al.2011] Manish Agarwal, Rakshit Shah, and Prashanth Mannem. 2011. Automatic question generation using discourse cues. In *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–9. Association for Computational Linguistics.
- [Ali et al.2010] Husam Ali, Yllias Chali, and Sadid A Hasan. 2010. Automation of question generation from sentences. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 58–67.
- [Auer et al.2007] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.
- [Blei et al.2003] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.
- [Bollacker et al.2008] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD*, pages 1247–1250. ACM.
- [Chen et al.2009] W Chen, G Aist, and J Mostow. 2009. Generating questions automatically from informational text. In *Proceedings of the 2nd Workshop on Question Generation*, pages 17–24.
- [Chen et al.2011] Mengen Chen, Xiaoming Jin, and Dou Shen. 2011. Short text classification improved by learning multi-granularity topics. In *IJ-CAI*, pages 1776–1781. Citeseer.
- [Curto et al.2012] Sérgio Curto, A Mendes, and Luisa Coheur. 2012. Question generation based on lexico-syntactic patterns learned from the web. *Dialogue & Discourse*, 3(2):147–175.
- [Heilman and Smith2010] Michael Heilman and Noah A Smith. 2010. Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617. Association for Computational Linguistics.
- [Labutov et al.2015] Igor Labutov, Sumit Basu, and Lucy Vanderwende. 2015. Deep questions without deep understanding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL-15)*, pages 889–898, Beijing, China.
- [Lindberg et al.2013] David Lindberg, Fred Popowich, John Nesbit, and Phil Winne. 2013. Generating natural language questions to support learning on-line. pages 105–114.
- [Liu and Zhang2015] Jiangming Liu and Yue Zhang. 2015. An empirical comparison between n-gram and syntactic language models for word ordering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP-15)*, pages 369–378, Lisbon, Portugal.
- [Ma et al.2015] Chenglong Ma, Weiqun Xu, Peijia Li, and Yonghong Yan. 2015. Distributional representations of words for short text classification. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 33–38, Denver, Colorado.
- [Mazidi and Nielsen2014] Karen Mazidi and Rodney D Nielsen. 2014. Linguistic considerations in automatic question generation. In *Proceedings of ACL*, pages 321–326.
- [Mikolov and Dean2013] T Mikolov and J Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*.
- [Olney et al.2012] Andrew M Olney, Arthur C Graesser, and Natalie K Person. 2012. Question generation from concept maps. *Dialogue and Discourse*, 3(2):75–99.
- [Papineni et al.2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Conference of the Association for Computational Linguistics (ACL-02)*, pages 311–318.
- [Phan et al.2008] Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. 2008. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th international conference on World Wide Web*, pages 91–100.
- [Rus et al.2010] Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Cristian Moldovan. 2010. Overview of the first question generation shared task evaluation challenge. In *Proceedings of the Third Workshop on Question Generation*, pages 45–57.
- [Serban et al.2016] Iulian Vlad Serban, Alberto García-Durán, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. 2016. Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL-16)*, pages 588–598, Berlin, Germany.
- [Seyler et al.2015] Dominic Seyler, Mohamed Yahya, and Klaus Berberich. 2015. Generating quiz questions from knowledge graphs. In *Proceedings of the 24th International Conference on World Wide Web*, pages 113–114. ACM.

[Wolfe1976] John H Wolfe. 1976. Automatic question generation from text-an aid to independent study. *ACM SIGCSE Bulletin*, 8(1):104–112.